GE.TRACKER: A ROBUST, LIGHTWEIGHT TOPIC TRACKING SYSTEM

Tomek Strzalkowski, Gees C. Stein and G. Bowden Wise

GE Corporate Research & Development
1 Research Circle
Niskayuna, NY 12309
strzalkowski@crd.ge.com
phone: 518-387-6871
fax: 518-387-6845

ABSTRACT

We describe a topic tracking system developed at GE R&D Center in connection with our participation in DARPA TDT evaluations. The TDT tracking task is specified as follows: given Nt training news stories on a topic, the system must find all subsequent stories on the same topic in all tracked news sources. These sources include radio and television news broadcasts, as well as newswire feeds. The initial set of training stories (usually 1, 2 or 4) is the only information about the topic available to the tracking system.[1] The tracking performance is gauged using the False Alarm Rate and the Miss Rate metrics, reflecting the incidence of incorrect classification decisions made by the automatic system.

1. OVERALL DESIGN

GE.Tracker has been developed in a course of a few weeks during the summer of 1998. It has been designed to achieve a reasonably high-accuracy performance using a lightweight, extremely portable and robust algorithms that rely on content compression rather than on corpus statistics to detect relevance and assess topicality of the source material. The tracker operates by first creating a topic tracking query (TQ) out of the available training stories. Subsequently, it accepts incoming stories, summarizes them topically, scores the summaries for content, then assesses content relevance to the tracking query. Stories whose compressed content summaries clear the empirically established threshold are classified as being "on topic".

2. BUILDING TOPIC TRACKING QUERY

The topic tracking query (TQ) is built out of available training material, which consists of Nt news stories on a topic of interest. In TDT2 evaluations the default value of Nt has been 4 topical stories. The initial tracking query TQ0 is formed out of the most frequent non-stop words and collocations found in the training set. The frequency threshold is set so that the words selected for the TQ0 occur at least Nt-1 times in the training set (the exact formula will be explained in the final paper). Collocations are all pairs of TQ0 terms that occur in 2 or more training stories. At this time, all within-the-story co-occurrences are collected. More advanced proximity or co-dependency calculations are planned for the future.

All terms and collocations are weighted to reflect their distribution within the training stories. Terms and collocations that occur in every training story are assigned greater weights than those occurring in fewer stories. The overall term frequency in the training set is factored in, but no external statistics are used. In particular, no collection level term weights, such as the inverted document frequency (idf) are computed.[2]

3. TRACKING INCOMING STORIES

The current version of GE.Tracker assumes that the input consists of a continuous broadcast stream pre-segmented into stories, although this segmentation does not have to be accurate. A future version is planned where an unsegmented broadcast stream can be used. The incoming stories are either text (newswire stories) or transcripts of audio tracks from television or radio news programs. These transcripts do not have to be accurate either, so the output of an automated speech recognition system is acceptable. Indeed we report here mostly on tracking results with automatically derived transcripts.

The incoming stories, whether text or speech transcripts, are summarized topically using the GE.Summarizer developed under the Tipster program. For TDT evaluations the summarizer has been modified to derive compressed content capsules of news stories, rather than true summaries. Since the capsules are not meant to be human-readable, we make no effort to maintain readability (there are no explicit paragraph or sentence boundaries available in automated broadcast transcripts). Furthermore, a tighter content compression is required than normally provided by the summarizer.
Summaries are scored for content density and topic coverage. The scoring method has been designed so that a summary obtains a high score only if it captures the dominating theme of the full story rather than some side aspect. Furthermore, only those high-scoring summaries that "cover" the tracking query TQ0 are selected as representing stories that are on topic. The concept of tracking query coverage is an attempt to quantify content distribution in a highly concentrated, yet naturally flowing news-style language. We use an empirically validated formula that requires specific coverage ratios depending upon the query length and term rankings (e.g., 45% for the 15 top-ranking terms in the query, 20% for the next 10 terms, etc.).

4. AUTOMATIC SUMMARIZATION

We summarize incoming stories in order to compress their content; specifically, we are only interested what a given story has to say on the topic being tracked. Currently, 5% topical summaries are derived. This may or may not be an optimal length, and indeed further experiments are needed to determine what the optimal summary length might be for each story. The basic summarization algorithm is outlined below. We present a simplified version of our original text summarization algorithm; the present version ignores all readability considerations we normally impose upon the summaries. This allows to produce shorter and tightly compressed summaries, while requiring no special provisions for dealing with continuous stream of words, with no paragraphs or sentence boundaries.

The summarizer can work in two modes: generic and topical. In the generic mode, it simply summarizes the main points of the original document. In the topical mode, it takes a user supplied statement of interest, a topic, and derives a summary related to this topic. A topical summary is thus usually different from the generic summary of the same document. The summarizer can produce both indicative and informative summaries. An indicative summary, typically 5-10% of the original text, is when there is just enough material retained from the original document to indicate its content. An informative summary, on the other hand, typically 20-30% of the text, retains all the relevant facts that a user may need from the original document, that is, it serves as a condensed surrogate, a digest. The summarization proceeds in the following steps:

1. Segment text into passages. Use any available handles, including indentation, SGML, empty lines, sentence ends, etc. If no paragraph or sentence structure is available, use approximately equal size chunks. In TDT evaluations, we used 33-word text chunks.

2. Build a paragraph-search query out of the content words, phrases and other terms found in the title, a user-supplied topic description (if available), as well as the terms occurring frequently in the text.
3. Score all passages with respect to the paragraph-search query. Assign a point for each co-occurring term. The goal is to maximize the overlap, so multiple occurrences of the same term do not increase the score.
4. Normalize passage scores by their length, taking into account the desired target length of the summary. The goal is to keep summary length as close to the target length as possible. The weighting formula is designed so that small deviations from the target length are acceptable, but large deviations will rapidly decrease the passage score. The exact formulation of this scheme depends upon the desired tradeoff between summary length and content. The following is the basic formula for scoring passage P of length l against the passage-search query Q and the target summary length of t:

$$\text{NormScore}(P,Q) = \{\text{RawScore}(P,Q)\}/\{\sqrt{\{|l-t|/t + 1\}}$$

$$\text{RawScore}(P,Q) = \sum_{q \in Q} \{\text{weight}(q,P)\}$$

with sum over unique content terms q, and unit weights.
5. Discard all passages with length in excess of 1.5 times the target length. This reduces the number of passage combinations the summarizer has to consider, thus improving its efficiency. The decision whether to use this condition depends upon our tolerance to length variability. In extreme cases, to prevent obtaining empty summaries, the summarizer will default to the first paragraph of the original text.
6. Combine passages into groups of 2 or more based on their content, composition and length. The goal is to maximize the score, while keeping the length as close to the target length as possible. Any combination of passages is allowed, including non-consecutive passages, although the original ordering of passages is retained.
7. Recalculate scores for all newly created groups. This is necessary, and cannot be obtained as a sum of scores because of possible term repetitions. Discard any passage groups longer than 1.5 times the target length.
8. Rank passage groups by score. All groups become candidate summaries.
9. Repeat steps 6 through 8 until there is no change in top-scoring passage group through 2 consecutive iterations. Select the top scoring passage or passage group as the final summary.

5. ADAPTIVE TRACKING

For tracking long-running topics that evolve over time, adaptive tracking is achieved through continuous modification of the tracking query, TQ0, TQ1, TQ2, etc. This is done by augmenting the initial set of training stories with selected topical stories in the broadcast stream and recomputing the tracking query. In TDT evaluations adaptive tracking has not been useful for most topics.

6. PRELIMINARY RESULTS

Preliminary results of dry run evaluation (8/98) show that GE.Tracker achieves 13% miss rate and 0.8% false alarm rate for the automated speech transcription input, and 11% and 0.6%, respectively for manual speech transcription. These results are already very close to the target values of 10% and 0.1% respectively. In the final TDT2 run we scored 14% miss rate and 1.9% false alarm rate for automated speech transcription conditions, and 13% and 1.4%, respectively, for manually transcribed sources (FDCH data). It may be worth noting that the final evaluation test was harder than the dry run tests, which is reflected in the increased FA rates. We are planning to perform a few more contrastive tests before the workshop.

7. EXTENSIONS

We plan to revise how we do tracking by changing the way we compute weights for terms. The revised process is as follows:

7.1. Training

Using the N_t training stories, we remove stop words and stem and calculate term frequencies for the remaining stems. We then form the training query TQ out of the most frequent stems found in training stories. Each stem tq_i is assigned a weight w_i, defined as:

$$w_i = tf_i / N_t$$

where tf_i is the term frequency of term i across all N_t stories. Note that this is a term frequency in TQ and does not consider term distribution across the N_t docs. We form the weighted term vector for TQ as TF = { w_i }.

7.2. Tracking

During tracking we summarize each story, and keep counts of how many stories have been read and also how many stories (summaries) each term in TQ has occurred in:

N = number of stories read thus far
DF = { df_i: number of stories (summaries) in which term i has occurred }

Using this information, we re-compute the weights for each term appearing in TQ as

$$wt_i = \log (N / df_i)$$

forming revised tracking query vector TFS = { wt_i }. For each incoming story S we calculate the tracking score. To arrive at a score for the story, we first form the story vector SF using the summary of the story, SF = { ws_i } where $ws_i = wt_i$ if term *i* occurs in the summary, and is set to 0 otherwise. We then compute the score as a cosine of the angle between the two vectors:

$$score(S) = \sum_{j \in SF} [(wt_j * ws_j) / (|TFS| * |SF|)]$$

7.3. Furthermore

The reader may notice that in this revised model the system learns as it tracks more stories, so its performance is expected to improve over time. In order to improve the initial performance, the tracker can be pre-trained on similar material before any tracking begins. In TDT-2 stories preceding the last training story for each topic could be so used. The more of the pre-training material is available, the better, but we may also note that it is required only when the tracker is used for the first time, or when it is restarted after a longer period of time. In general, pre-training on old news (a few months old) is likely to be less effective than on more recent material.

Note also that our training method does not distinguish between on-topic and off-topic stories while gathering term statistics. This extra information was available (nominally) in TDT-2, but may not be available in TDT-3. In general, adaptive training using positive and negative examples, as opposed to overall distribution statistics, should converge faster and produce better tracking queries, as evidenced by relevance feedback methods used in document retrieval. However, it is unrealistic to expect that appropriately varied set of samples will be available in news tracking applications.

REFERENCES

Strzalkowski, T., G. Stein, B. Wise. 1998. Robust, Practical Text Summarization. To appear in M. Maybury & I. Mani (eds.). Advances in Text Summarization.

Strzalkowski, T., G. Stein, B. Wise. 1998. Natural Language Information Retrieval: TREC-7 Report. Proceedings of the 7th Text Retrieval Conference (TREC-7). NIST.

[1] In the recently completed TDT2 evaluation, a large (100's) set of non-topical stories were also available. These were the stories reported before and in-between the topical training stories in continuous broadcast streams. This additional information could be used in training the tracking system, although this made the test slightly less realistic. The GE Tracker does not use the non-topical training stories.

[2] Determining "typical" term distribution within news stories is useful in order to eliminate or "downgrade" terms that are relatively common for many unrelated stories. In addition to the usual list of stopwords (such as determiners, prepositions, and various pro-forms) we identify and eliminate a certain number of words commonly used in news reporting: "said", "today", "time", etc.